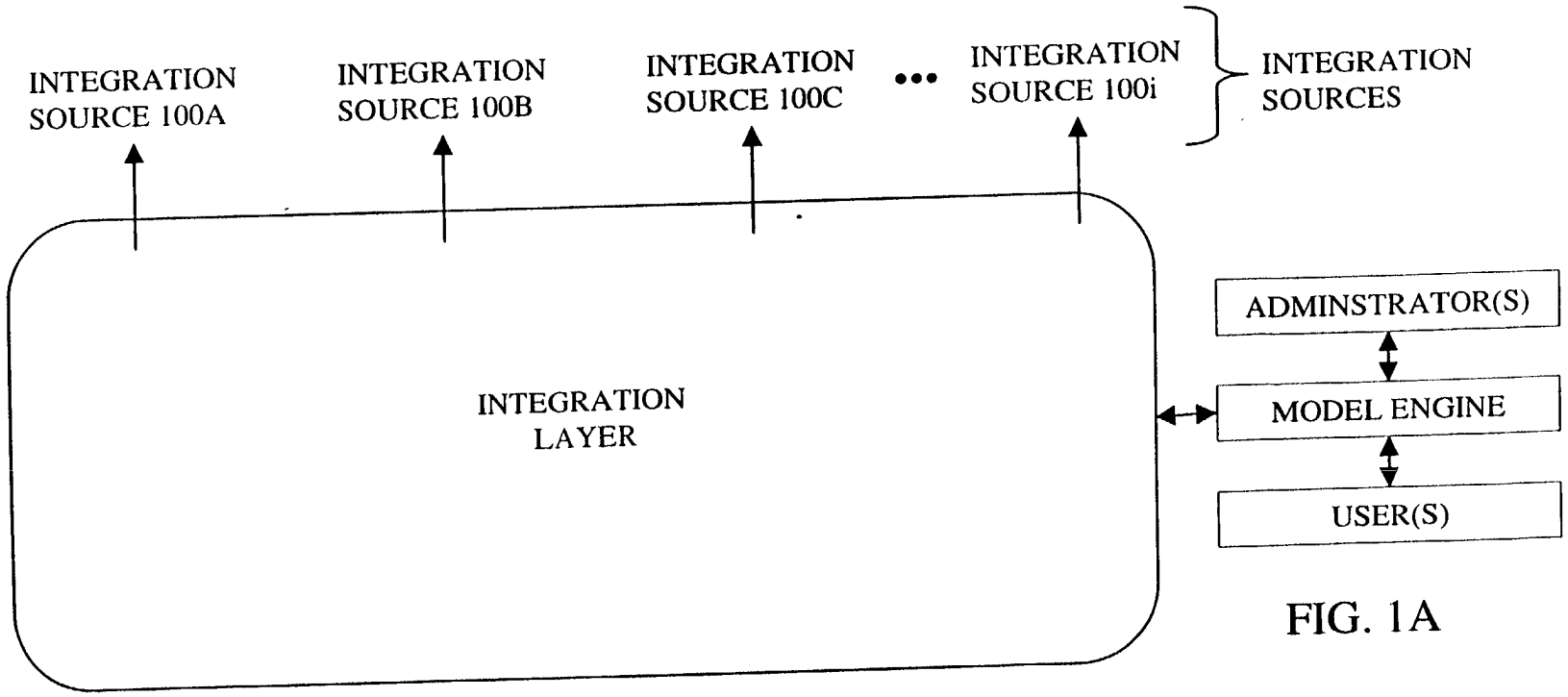
INTEGRATION SOURCE 100A
INTEGRATION SOURCE 100B
INTEGRATION SOURCE 100C
INTEGRATION SOURCE 100i
INTEGRATION SOURCES
INTEGRATION LAYER
ADMINSTRATOR(S)
MODEL ENGINE
USER(S)

FIG. 1A

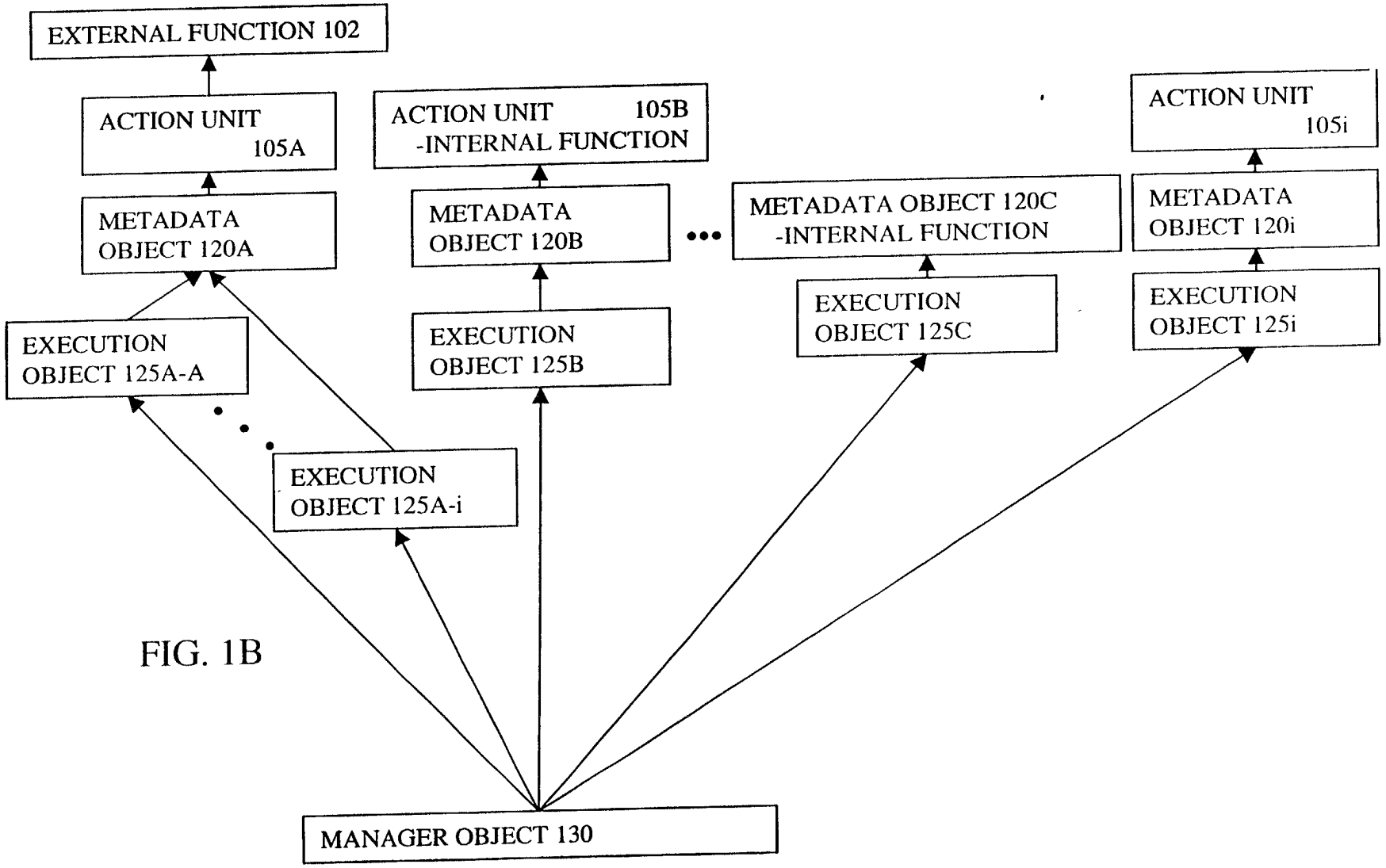
EXTERNAL FUNCTION 102
ACTION UNIT 105A
ACTION UNIT 105B -INTERNAL FUNCTION
ACTION UNIT 105i
METADATA OBJECT 120A
METADATA OBJECT 120B
METADATA OBJECT 120C -INTERNAL FUNCTION
METADATA OBJECT 120i
EXECUTION OBJECT 125A-A
EXECUTION OBJECT 125A-i
EXECUTION OBJECT 125B
EXECUTION OBJECT 125C
EXECUTION OBJECT 125i
MANAGER OBJECT 130

FIG. 1B

TOSTOT" 6TE/TOOT

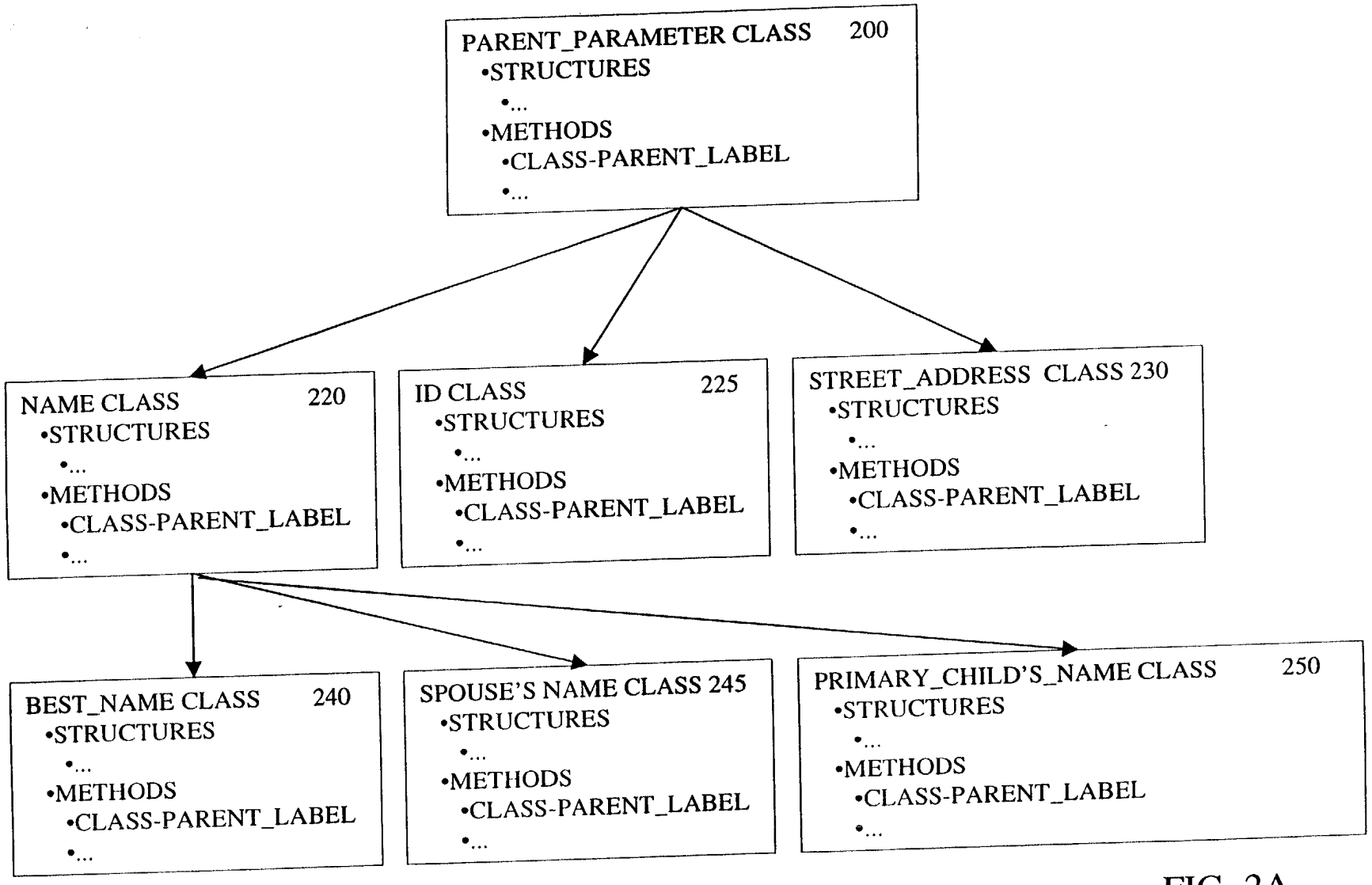

FIG. 2A

KEY_VALUE CLASS 205
- •STRUCTURES
  - •KEY_STORAGE <KEY OR POINTER TO A KEY>
  - •VALUE_STORAGE <VALUE OR PTR>
  - •...
- •METHODS
  - •...

FIG. 2B

REGISTER CLASS 210
- •STRUCTURES
  - •ACTION_TRACKING <COLLECTION OF KEY_VALUE OBJECTS>
  - •...
- •METHODS
  - •REGISTER_ACTION (<KEY>; <PTR TO AN ACTION UNIT>)
  - •GET_ACTION (<KEY>)
  - •...

FIG. 2C

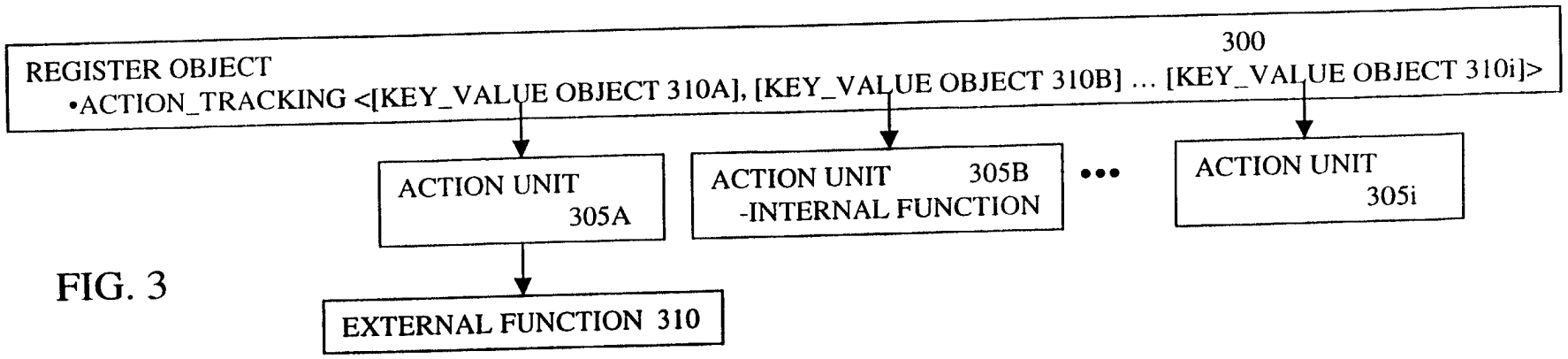

FIG. 3

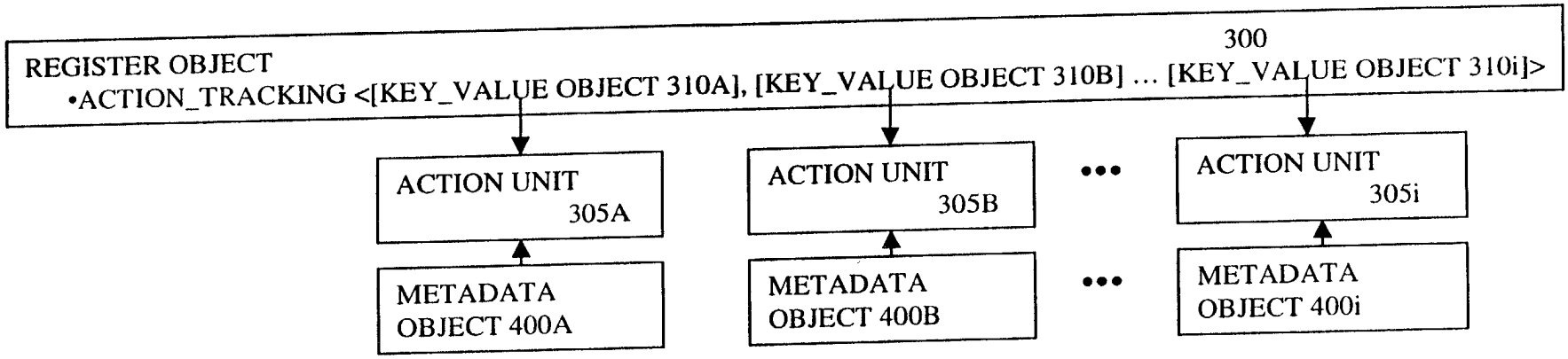
REGISTER OBJECT
•ACTION_TRACKING <[KEY_VALUE OBJECT 310A], [KEY_VALUE OBJECT 310B] ... [KEY_VALUE OBJECT 310i]>
300
ACTION UNIT 305A
ACTION UNIT 305B
ACTION UNIT 305i
METADATA OBJECT 400A
METADATA OBJECT 400B
METADATA OBJECT 400i

FIG. 4

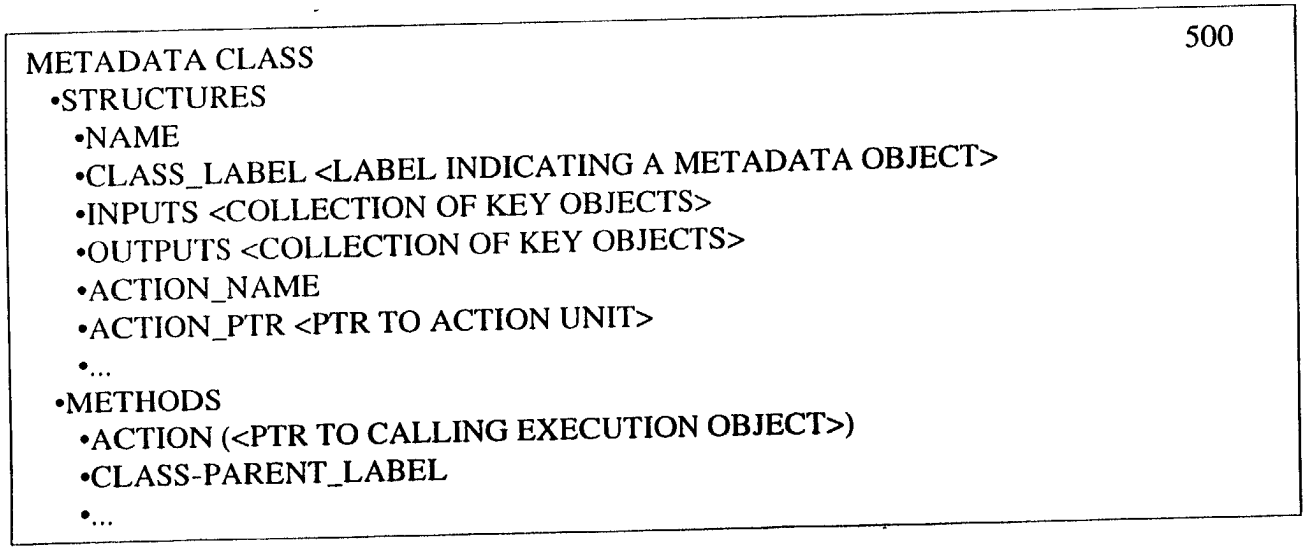
500
METADATA CLASS
•STRUCTURES
•NAME
•CLASS_LABEL <LABEL INDICATING A METADATA OBJECT>
•INPUTS <COLLECTION OF KEY OBJECTS>
•OUTPUTS <COLLECTION OF KEY OBJECTS>
•ACTION_NAME
•ACTION_PTR <PTR TO ACTION UNIT>
•...
•METHODS
•ACTION (<PTR TO CALLING EXECUTION OBJECT>)
•CLASS-PARENT_LABEL
•...

FIG. 5

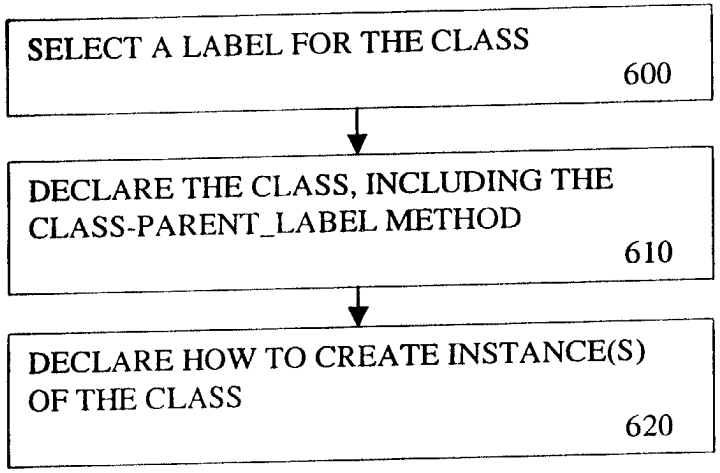

FIG. 6A

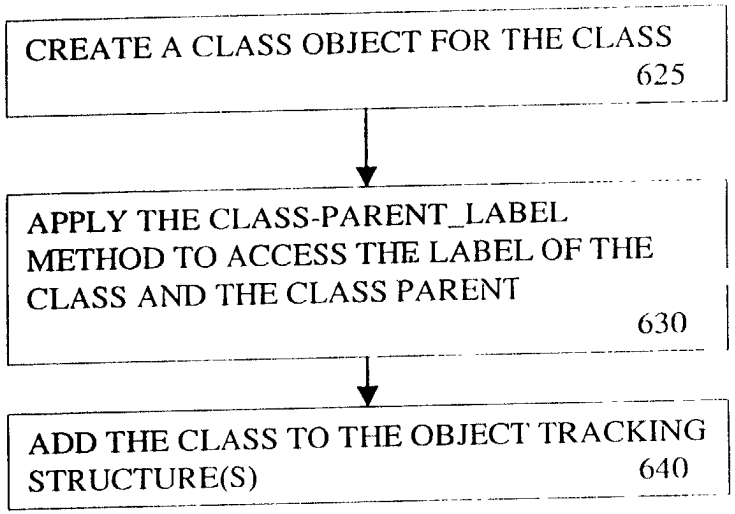

FIG. 6B

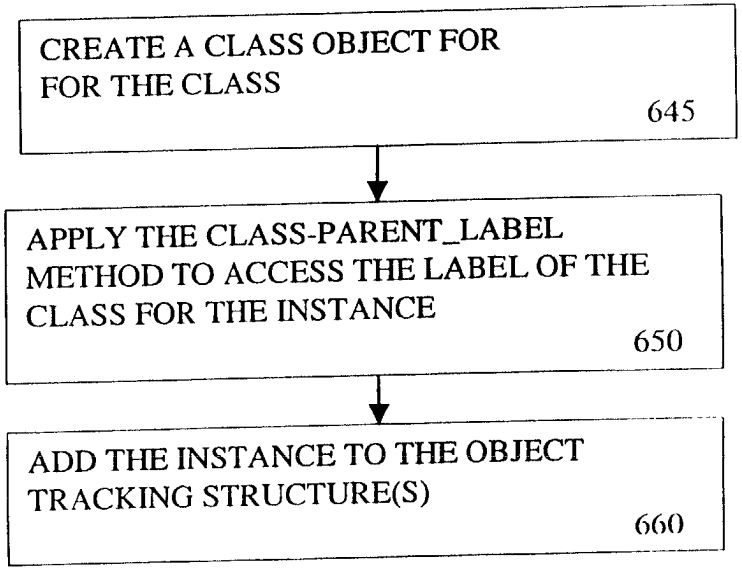

FIG. 6C

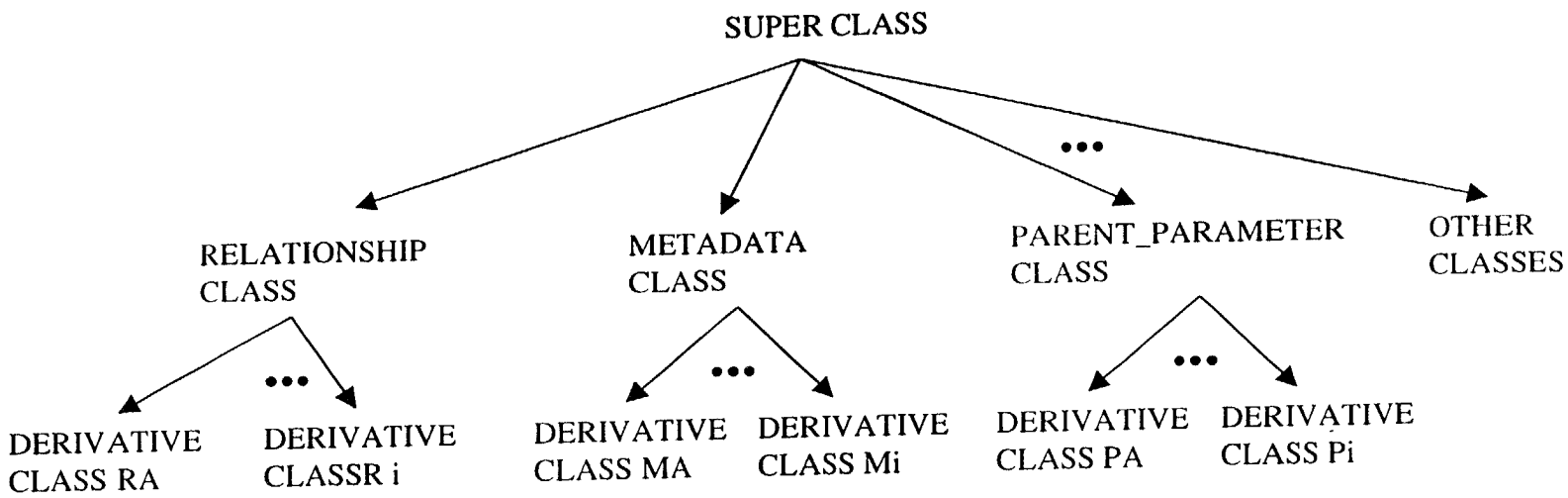

FIG. 7

SUBCLASS LABEL	NAME(S) OF INSTANCE(S)
RELATIONSHIP CLASS	<NAME1>, <NAME2> ...
TRACKING CLASS	
PARENT_PARAMETER CLASS	
DERIVATIVE CLASS PA	
DERIVATIVE CLASS Pi	
.	
.	
.	

FIG. 8

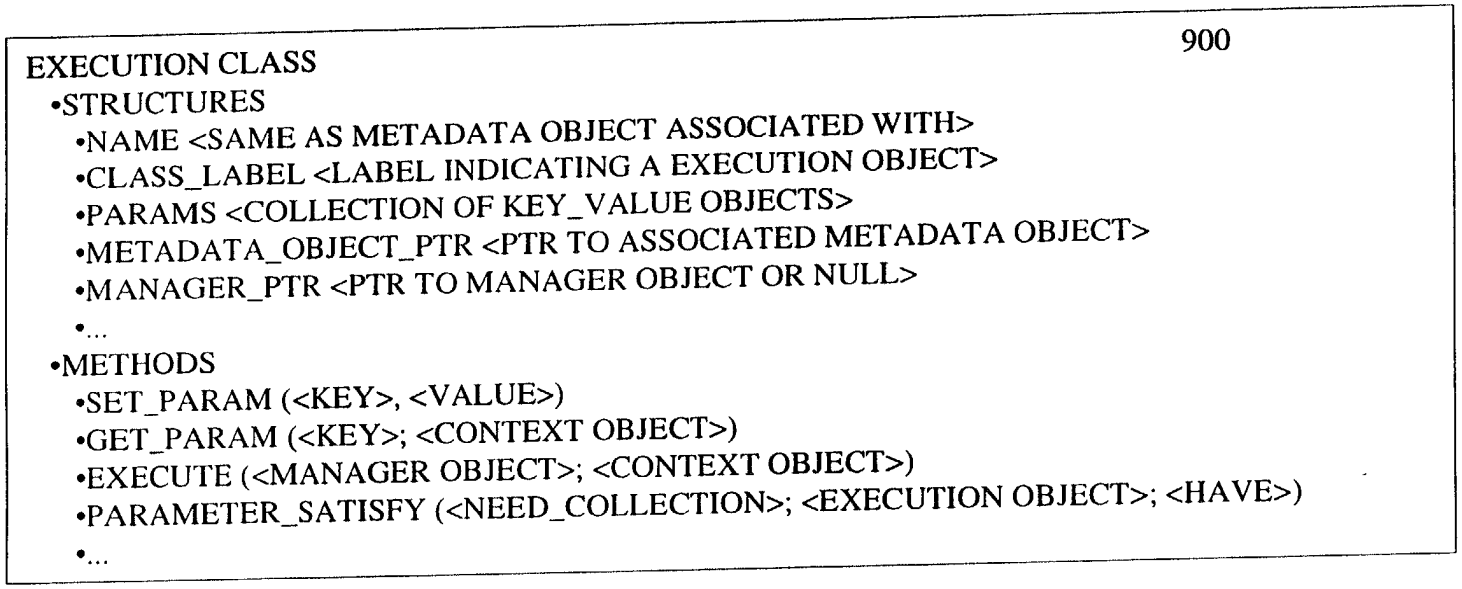
900
EXECUTION CLASS
•STRUCTURES
•NAME <SAME AS METADATA OBJECT ASSOCIATED WITH>
•CLASS_LABEL <LABEL INDICATING A EXECUTION OBJECT>
•PARAMS <COLLECTION OF KEY_VALUE OBJECTS>
•METADATA_OBJECT_PTR <PTR TO ASSOCIATED METADATA OBJECT>
•MANAGER_PTR <PTR TO MANAGER OBJECT OR NULL>
•...
•METHODS
•SET_PARAM (<KEY>, <VALUE>)
•GET_PARAM (<KEY>; <CONTEXT OBJECT>)
•EXECUTE (<MANAGER OBJECT>; <CONTEXT OBJECT>)
•PARAMETER_SATISFY (<NEED_COLLECTION>; <EXECUTION OBJECT>; <HAVE>)
•...

FIG. 9

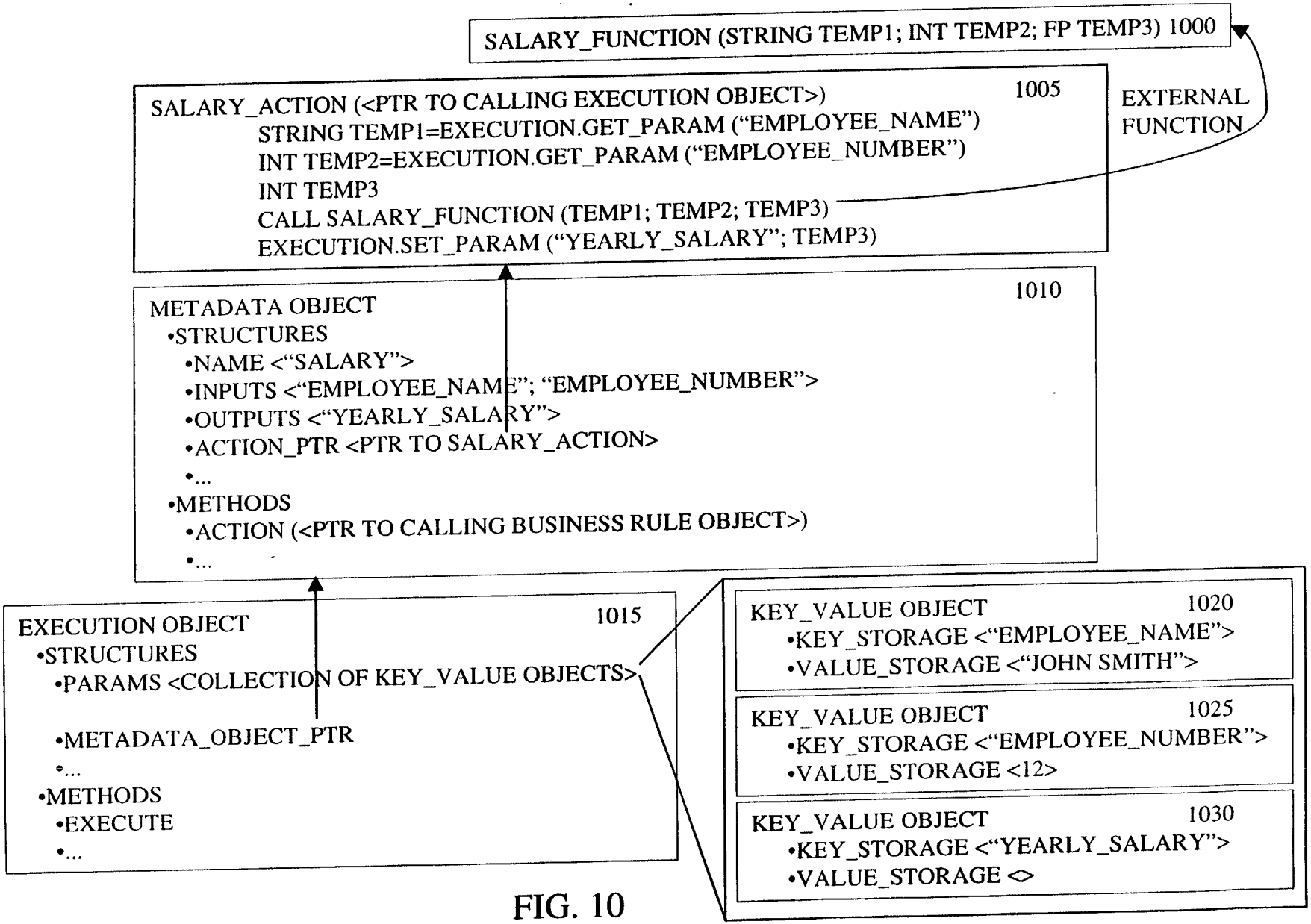
SALARY_FUNCTION (STRING TEMP1; INT TEMP2; FP TEMP3) 1000
SALARY_ACTION (<PTR TO CALLING EXECUTION OBJECT>) 1005
STRING TEMP1=EXECUTION.GET_PARAM ("EMPLOYEE_NAME")
INT TEMP2=EXECUTION.GET_PARAM ("EMPLOYEE_NUMBER")
INT TEMP3
CALL SALARY_FUNCTION (TEMP1; TEMP2; TEMP3)
EXECUTION.SET_PARAM ("YEARLY_SALARY"; TEMP3)
EXTERNAL FUNCTION
METADATA OBJECT 1010
•STRUCTURES
•NAME <"SALARY">
•INPUTS <"EMPLOYEE_NAME"; "EMPLOYEE_NUMBER">
•OUTPUTS <"YEARLY_SALARY">
•ACTION_PTR <PTR TO SALARY_ACTION>
•...
•METHODS
•ACTION (<PTR TO CALLING BUSINESS RULE OBJECT>)
•...
EXECUTION OBJECT 1015
•STRUCTURES
•PARAMS <COLLECTION OF KEY_VALUE OBJECTS>
•METADATA_OBJECT_PTR
•...
•METHODS
•EXECUTE
•...
KEY_VALUE OBJECT 1020
•KEY_STORAGE <"EMPLOYEE_NAME">
•VALUE_STORAGE <"JOHN SMITH">
KEY_VALUE OBJECT 1025
•KEY_STORAGE <"EMPLOYEE_NUMBER">
•VALUE_STORAGE <12>
KEY_VALUE OBJECT 1030
•KEY_STORAGE <"YEARLY_SALARY">
•VALUE_STORAGE <>

FIG. 10

1100

CONTEXT CLASS
- •STRUCTURES
  - •NAME
  - •KEY_VALUE_COLLECTION <COLLECTION OF KEY_VALUE OBJECTS>
  - •EXECUTION_COLLECTION <COLLECTION OF EXECUTION OBJECTS>
  - •...
- •METHODS
  - •...

FIG. 11A

1110

MANAGER CLASS
- •STRUCTURES
  - •CONTEXT_COLLECTION <COLLECTION OF CONTEXT OBJECTS>
  - •DEFAULT_CONTEXT <IDENTIFIES ONE OF CONTEXT OBJECTS IN CONTEXT_COLLECTION>
  - •EXECUTION_COLLECTION <COLLECTION OF EXECUTION OBJECTS MANAGED BY A MANAGER OBJECT>
  - •EXECUTION_PATH_COLLECTION <ORDERED COLLECTION OF LEVEL OBJECTS>
  - •...
- •METHODS
  - •NEW_EXECUTION (<NAME OF METADATA OBJECT>; <CLASS_LABEL>; <CONTEXT OBJECT>)
  - •FIND_BY_NAME (<NAME OF METADATA OBJECT>; < CLASS_LABEL >; <CONTEXT OBJECT>)
  - •FIND_BY_NEED (<NEED_COLLECTION (E.G., COLLECTION OF PARAMETER KEYS)>; <HAVE (E.G., CONTEXT OBJECT CONTAINING PARAMETERS HAVE VALUES FOR)>)
  - •FIND_BY_NAME_AND_EXECUTE (<NAME OF METADATA OBJECT>; < CLASS_LABEL >; <CONTEXT OBJECT>)
  - •FIND_BY_NEED_AND _EXECUTE (<NEED_COLLECTION (E.G., COLLECTION OF PARAMETER KEYS)>; <HAVE (E.G., CONTEXT OBJECT CONTAINING PARAMETERS HAVE VALUES FOR)>)
  - •...

FIG. 11B

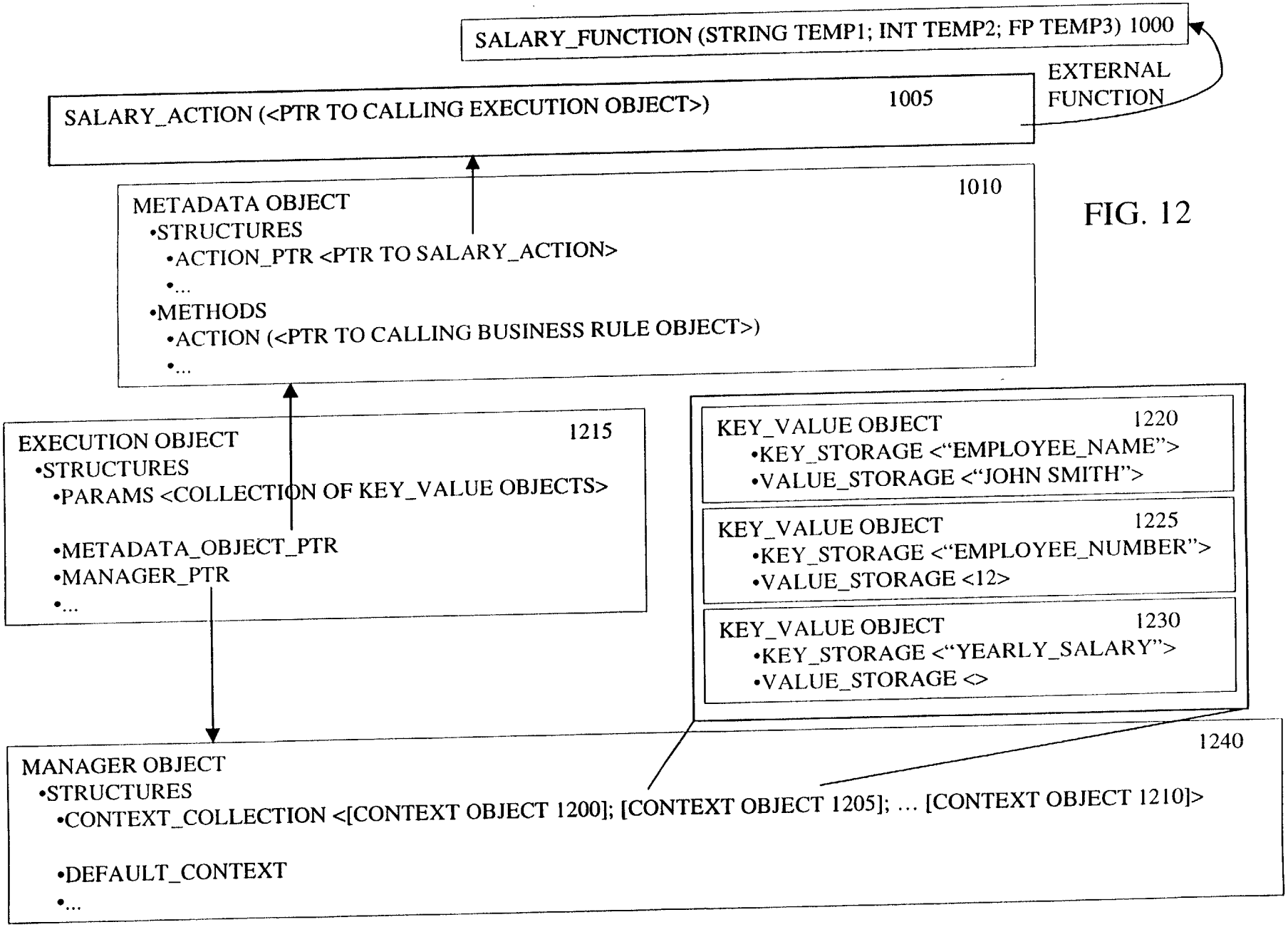
SALARY_FUNCTION (STRING TEMP1; INT TEMP2; FP TEMP3) 1000
EXTERNAL FUNCTION
SALARY_ACTION (<PTR TO CALLING EXECUTION OBJECT>) 1005
METADATA OBJECT 1010
•STRUCTURES
•ACTION_PTR <PTR TO SALARY_ACTION>
•...
•METHODS
•ACTION (<PTR TO CALLING BUSINESS RULE OBJECT>)
•...
EXECUTION OBJECT 1215
•STRUCTURES
•PARAMS <COLLECTION OF KEY_VALUE OBJECTS>
•METADATA_OBJECT_PTR
•MANAGER_PTR
•...
KEY_VALUE OBJECT 1220
•KEY_STORAGE <"EMPLOYEE_NAME">
•VALUE_STORAGE <"JOHN SMITH">
KEY_VALUE OBJECT 1225
•KEY_STORAGE <"EMPLOYEE_NUMBER">
•VALUE_STORAGE <12>
KEY_VALUE OBJECT 1230
•KEY_STORAGE <"YEARLY_SALARY">
•VALUE_STORAGE <>
MANAGER OBJECT 1240
•STRUCTURES
•CONTEXT_COLLECTION <[CONTEXT OBJECT 1200]; [CONTEXT OBJECT 1205]; ... [CONTEXT OBJECT 1210]>
•DEFAULT_CONTEXT
•...

FIG. 12

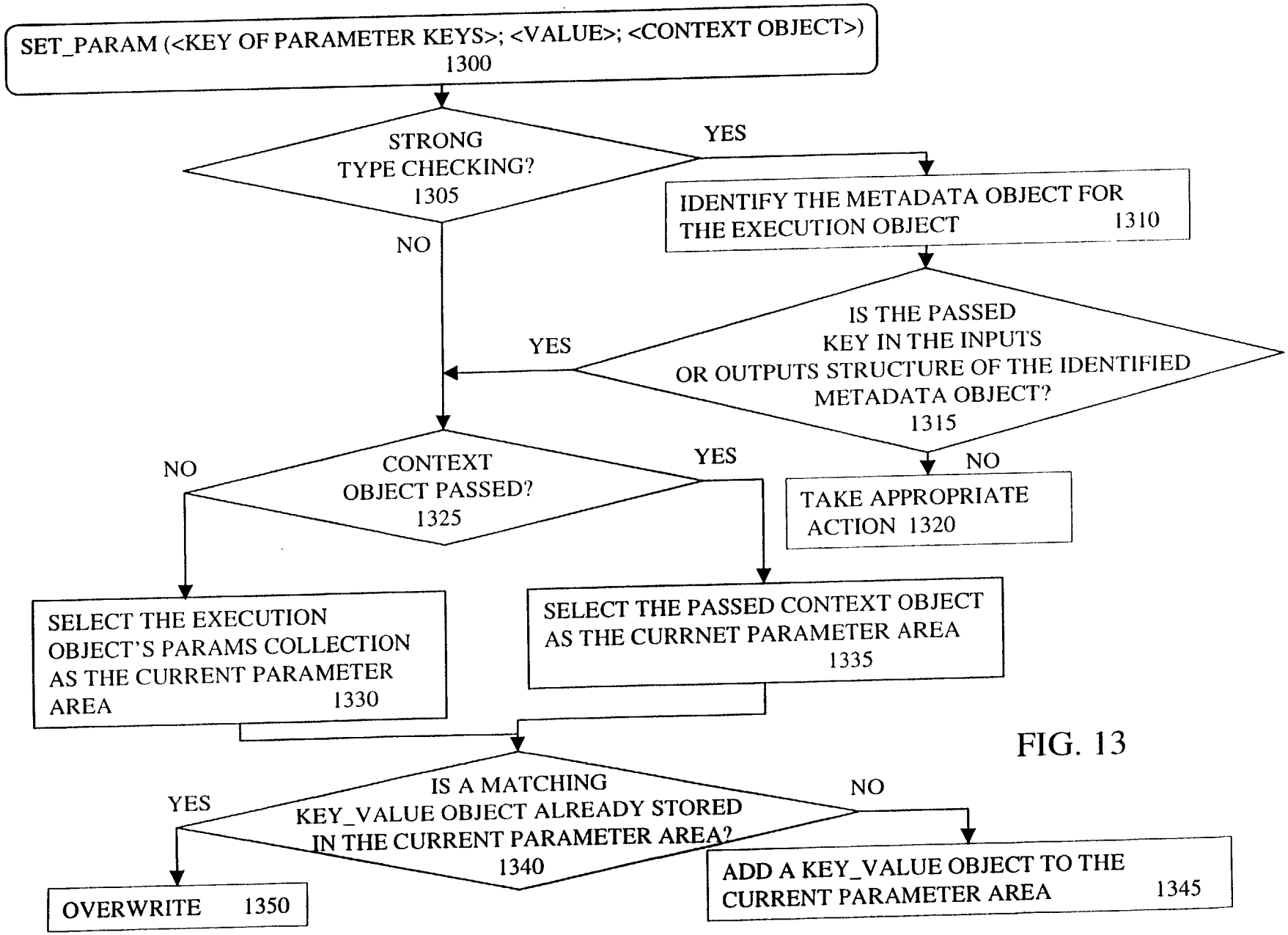
SET_PARAM (<KEY OF PARAMETER KEYS>; <VALUE>; <CONTEXT OBJECT>)
1300
STRONG TYPE CHECKING?
1305
YES
NO
IDENTIFY THE METADATA OBJECT FOR THE EXECUTION OBJECT 1310
IS THE PASSED KEY IN THE INPUTS OR OUTPUTS STRUCTURE OF THE IDENTIFIED METADATA OBJECT?
1315
YES
NO
TAKE APPROPRIATE ACTION 1320
CONTEXT OBJECT PASSED?
1325
NO
YES
SELECT THE EXECUTION OBJECT'S PARAMS COLLECTION AS THE CURRENT PARAMETER AREA 1330
SELECT THE PASSED CONTEXT OBJECT AS THE CURRNET PARAMETER AREA 1335
IS A MATCHING KEY_VALUE OBJECT ALREADY STORED IN THE CURRENT PARAMETER AREA?
1340
YES
NO
OVERWRITE 1350
ADD A KEY_VALUE OBJECT TO THE CURRENT PARAMETER AREA 1345

FIG. 13

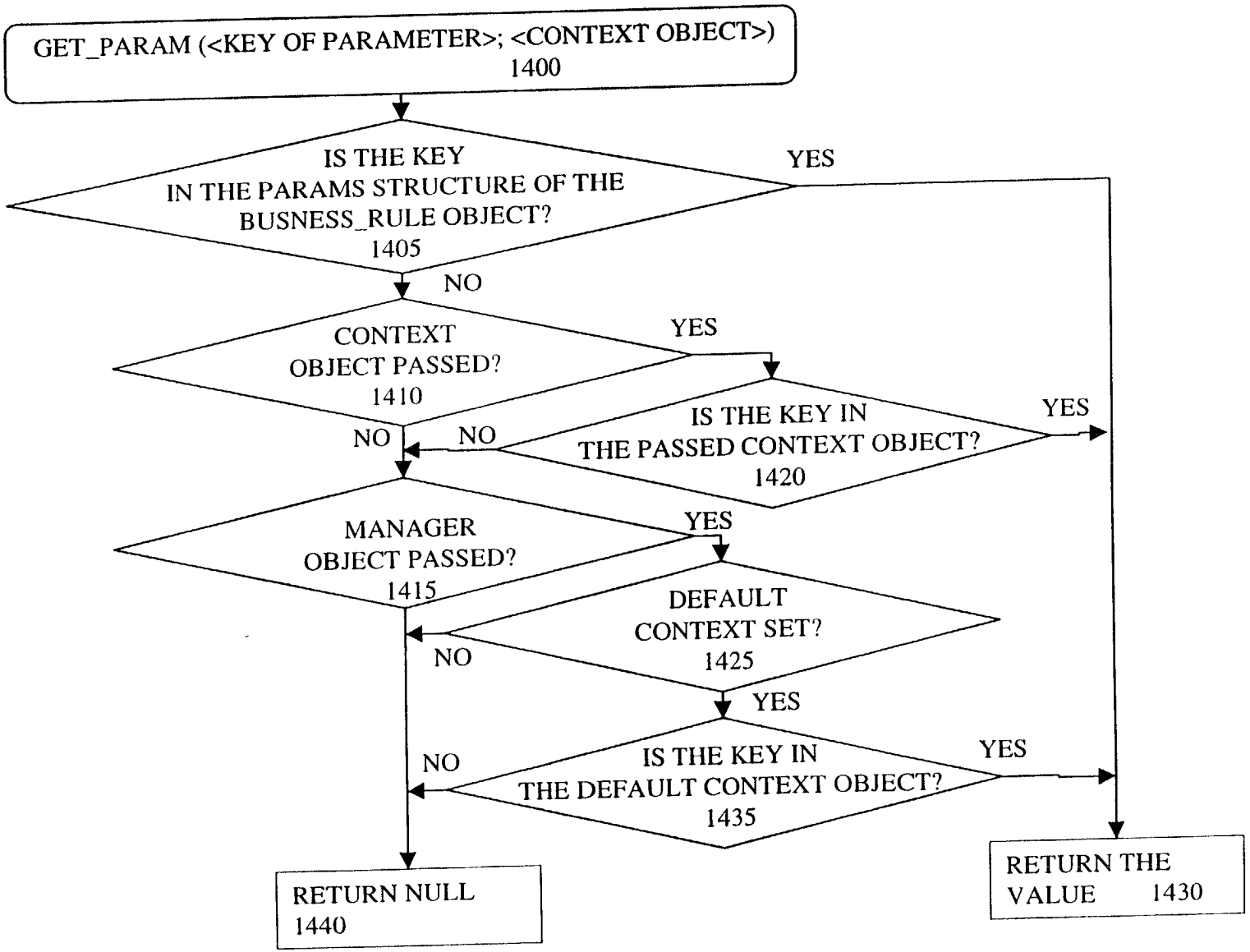
GET_PARAM (<KEY OF PARAMETER>; <CONTEXT OBJECT>)
1400
IS THE KEY IN THE PARAMS STRUCTURE OF THE BUSNESS_RULE OBJECT?
1405
YES
NO
CONTEXT OBJECT PASSED?
1410
YES
NO
IS THE KEY IN THE PASSED CONTEXT OBJECT?
1420
YES
NO
MANAGER OBJECT PASSED?
1415
YES
DEFAULT CONTEXT SET?
1425
NO
YES
IS THE KEY IN THE DEFAULT CONTEXT OBJECT?
1435
YES
NO
RETURN NULL
1440
RETURN THE VALUE 1430

FIG. 14

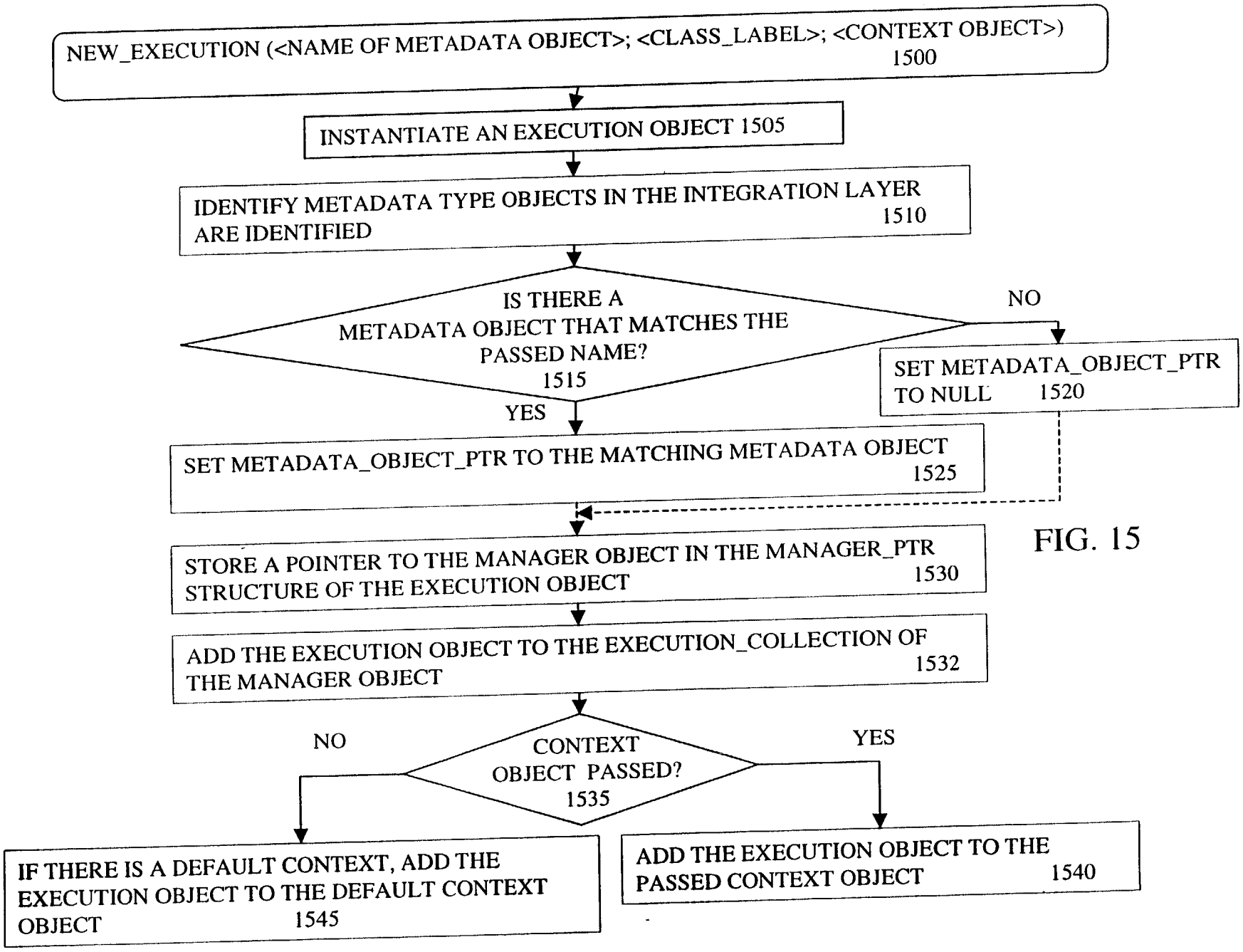
NEW_EXECUTION (<NAME OF METADATA OBJECT>; <CLASS_LABEL>; <CONTEXT OBJECT>)
1500
INSTANTIATE AN EXECUTION OBJECT 1505
IDENTIFY METADATA TYPE OBJECTS IN THE INTEGRATION LAYER ARE IDENTIFIED 1510
IS THERE A METADATA OBJECT THAT MATCHES THE PASSED NAME? 1515
NO
SET METADATA_OBJECT_PTR TO NULL 1520
YES
SET METADATA_OBJECT_PTR TO THE MATCHING METADATA OBJECT 1525
STORE A POINTER TO THE MANAGER OBJECT IN THE MANAGER_PTR STRUCTURE OF THE EXECUTION OBJECT 1530
ADD THE EXECUTION OBJECT TO THE EXECUTION_COLLECTION OF THE MANAGER OBJECT 1532
NO
CONTEXT OBJECT PASSED? 1535
YES
IF THERE IS A DEFAULT CONTEXT, ADD THE EXECUTION OBJECT TO THE DEFAULT CONTEXT OBJECT 1545
ADD THE EXECUTION OBJECT TO THE PASSED CONTEXT OBJECT 1540

FIG. 15

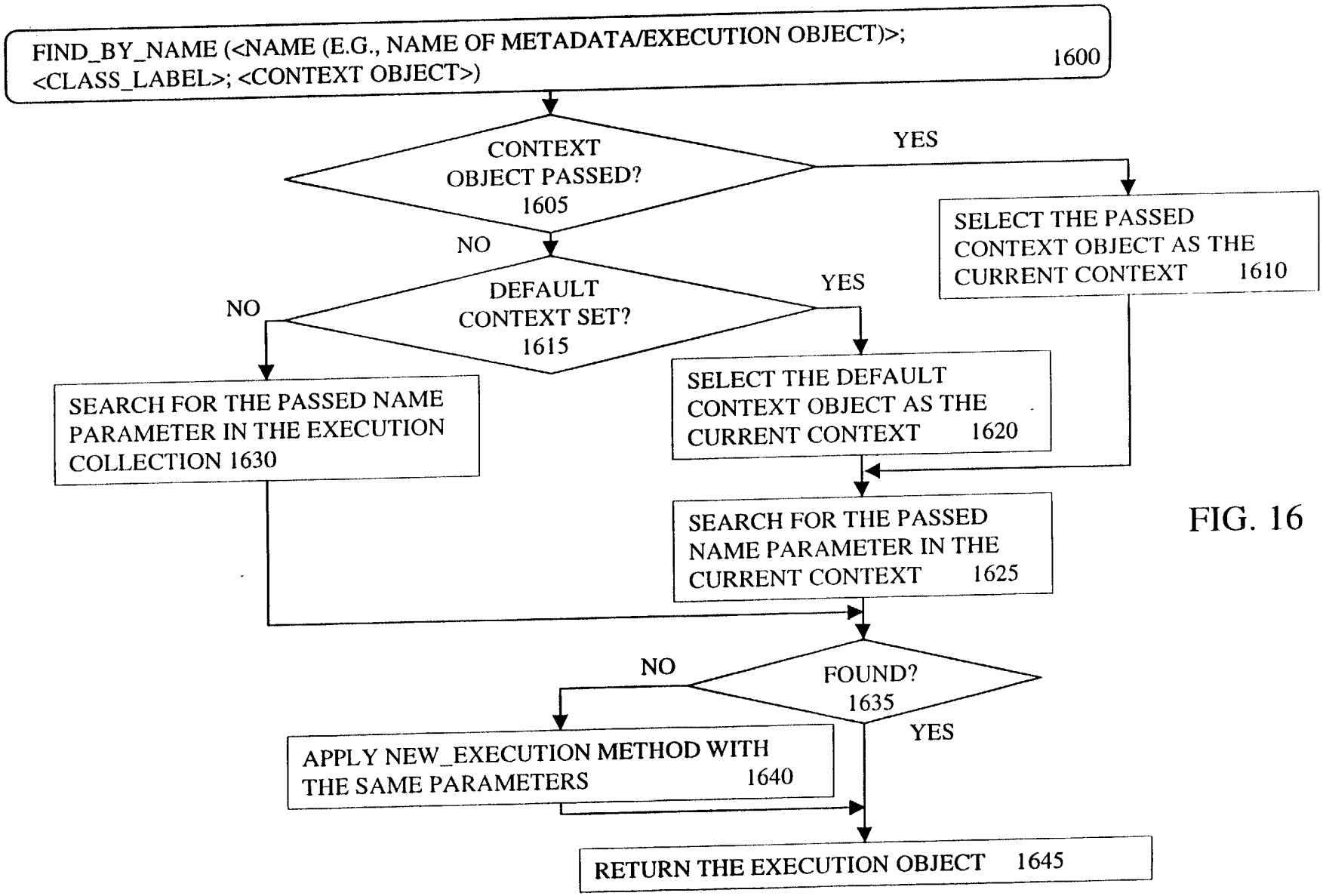
FIND_BY_NAME (<NAME (E.G., NAME OF METADATA/EXECUTION OBJECT)>; <CLASS_LABEL>; <CONTEXT OBJECT>)
1600
CONTEXT OBJECT PASSED?
1605
YES
SELECT THE PASSED CONTEXT OBJECT AS THE CURRENT CONTEXT 1610
NO
DEFAULT CONTEXT SET?
1615
YES
NO
SEARCH FOR THE PASSED NAME PARAMETER IN THE EXECUTION COLLECTION 1630
SELECT THE DEFAULT CONTEXT OBJECT AS THE CURRENT CONTEXT 1620
SEARCH FOR THE PASSED NAME PARAMETER IN THE CURRENT CONTEXT 1625
FOUND?
1635
NO
YES
APPLY NEW_EXECUTION METHOD WITH THE SAME PARAMETERS 1640
RETURN THE EXECUTION OBJECT 1645

FIG. 16

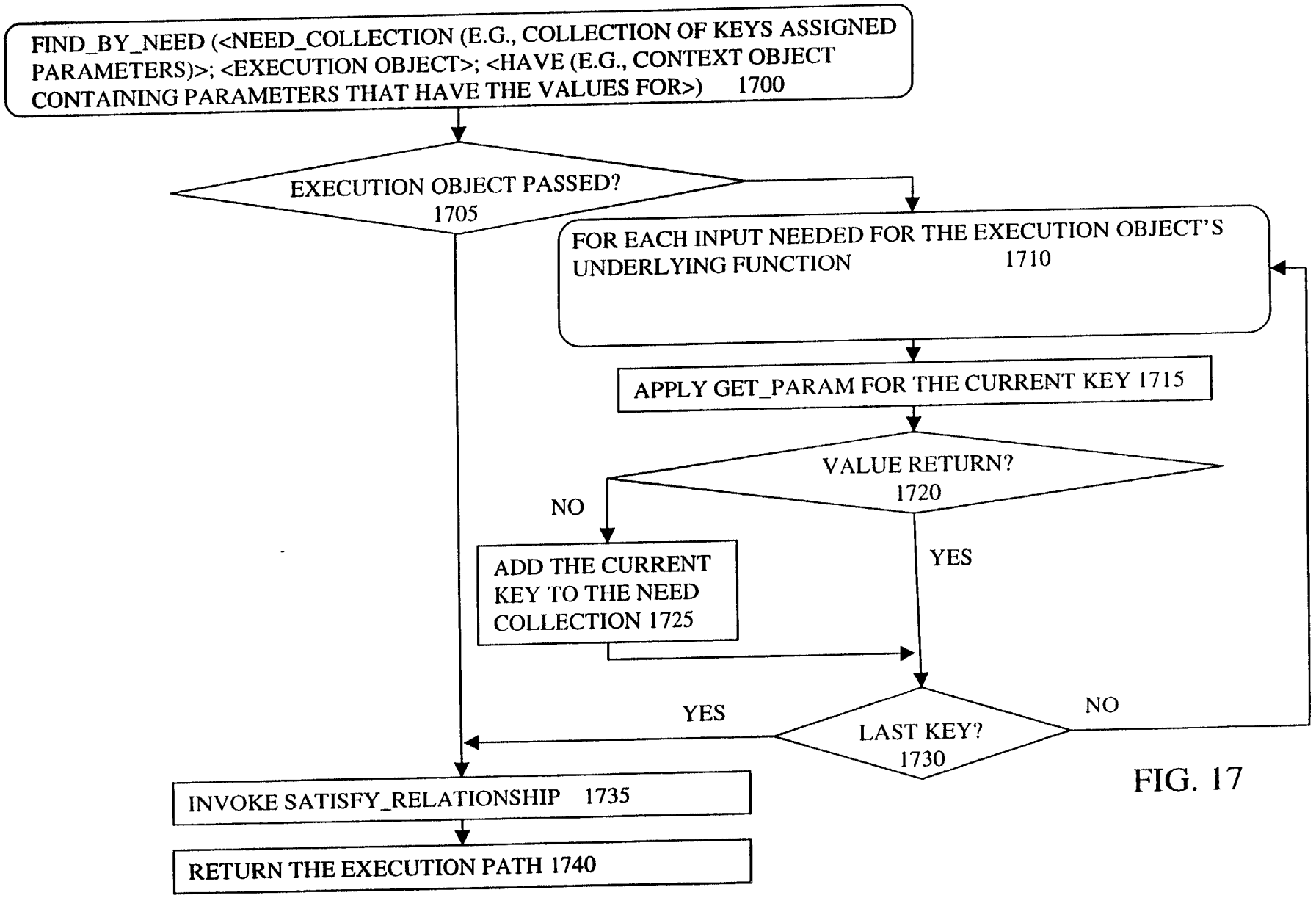
FIND_BY_NEED (<NEED_COLLECTION (E.G., COLLECTION OF KEYS ASSIGNED PARAMETERS)>; <EXECUTION OBJECT>; <HAVE (E.G., CONTEXT OBJECT CONTAINING PARAMETERS THAT HAVE THE VALUES FOR>) 1700
EXECUTION OBJECT PASSED? 1705
FOR EACH INPUT NEEDED FOR THE EXECUTION OBJECT'S UNDERLYING FUNCTION 1710
APPLY GET_PARAM FOR THE CURRENT KEY 1715
VALUE RETURN? 1720
NO
YES
ADD THE CURRENT KEY TO THE NEED COLLECTION 1725
LAST KEY? 1730
YES
NO
INVOKE SATISFY_RELATIONSHIP 1735
RETURN THE EXECUTION PATH 1740

FIG. 17

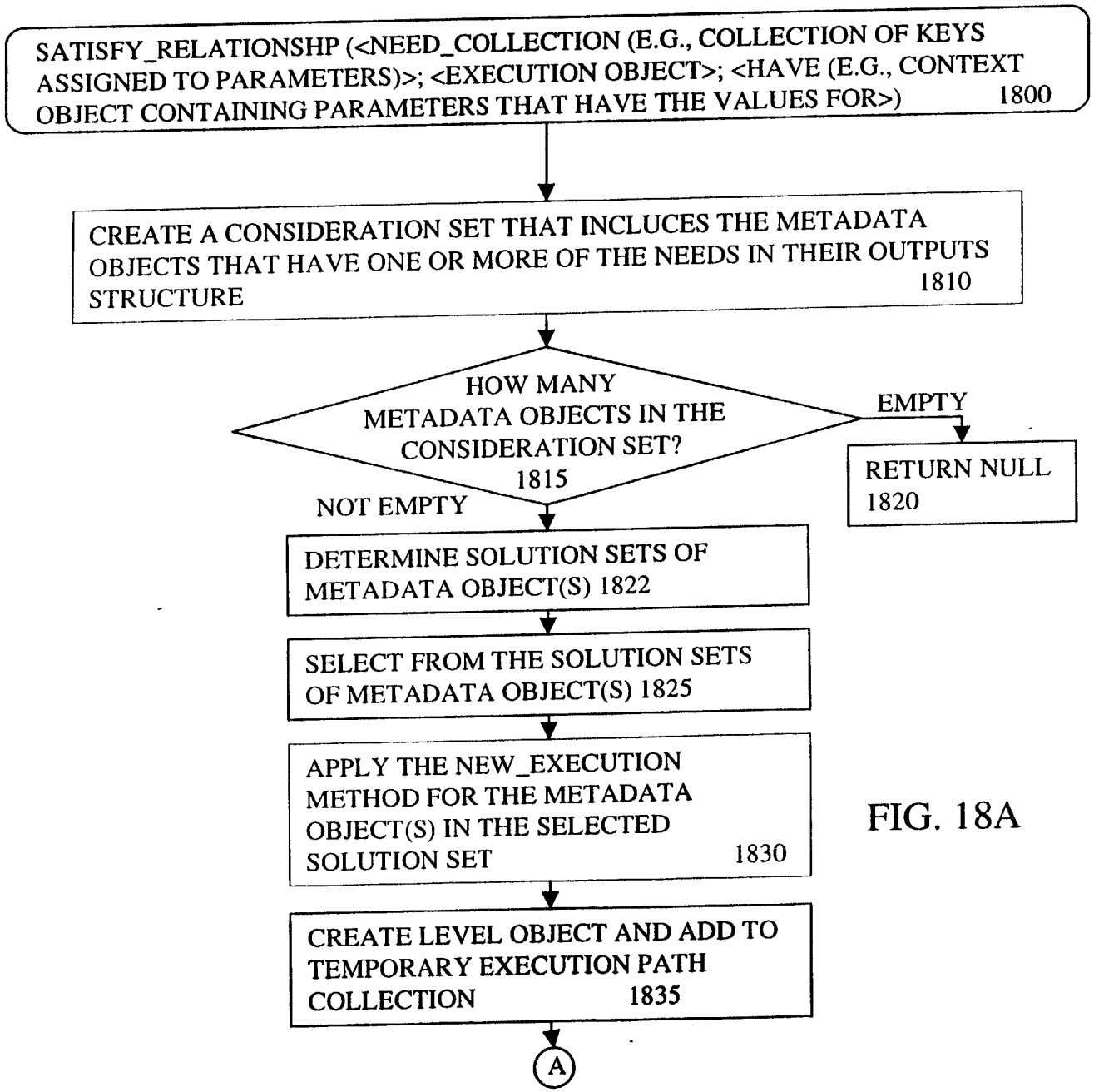
SATISFY_RELATIONSHP (<NEED_COLLECTION (E.G., COLLECTION OF KEYS ASSIGNED TO PARAMETERS)>; <EXECUTION OBJECT>; <HAVE (E.G., CONTEXT OBJECT CONTAINING PARAMETERS THAT HAVE THE VALUES FOR>) 1800
CREATE A CONSIDERATION SET THAT INCLUCES THE METADATA OBJECTS THAT HAVE ONE OR MORE OF THE NEEDS IN THEIR OUTPUTS STRUCTURE 1810
HOW MANY METADATA OBJECTS IN THE CONSIDERATION SET? 1815
EMPTY
RETURN NULL 1820
NOT EMPTY
DETERMINE SOLUTION SETS OF METADATA OBJECT(S) 1822
SELECT FROM THE SOLUTION SETS OF METADATA OBJECT(S) 1825
APPLY THE NEW_EXECUTION METHOD FOR THE METADATA OBJECT(S) IN THE SELECTED SOLUTION SET 1830
CREATE LEVEL OBJECT AND ADD TO TEMPORARY EXECUTION PATH COLLECTION 1835
A

FIG. 18A

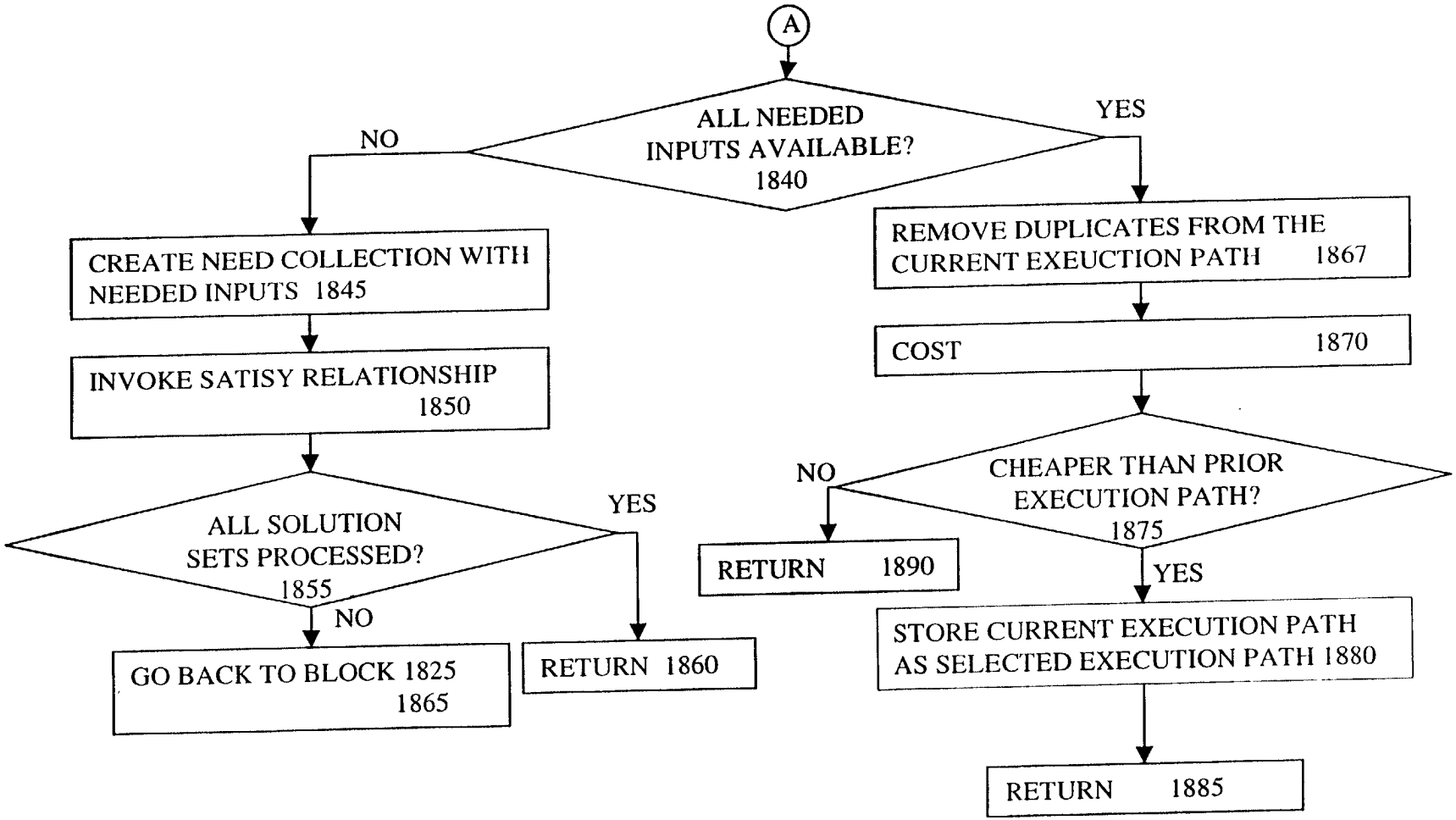
A
ALL NEEDED INPUTS AVAILABLE? 1840
NO
YES
CREATE NEED COLLECTION WITH NEEDED INPUTS 1845
INVOKE SATISY RELATIONSHIP 1850
ALL SOLUTION SETS PROCESSED? 1855
NO
YES
GO BACK TO BLOCK 1825 1865
RETURN 1860
REMOVE DUPLICATES FROM THE CURRENT EXEUCTION PATH 1867
COST 1870
CHEAPER THAN PRIOR EXECUTION PATH? 1875
NO
YES
RETURN 1890
STORE CURRENT EXECUTION PATH AS SELECTED EXECUTION PATH 1880
RETURN 1885

FIG. 18B

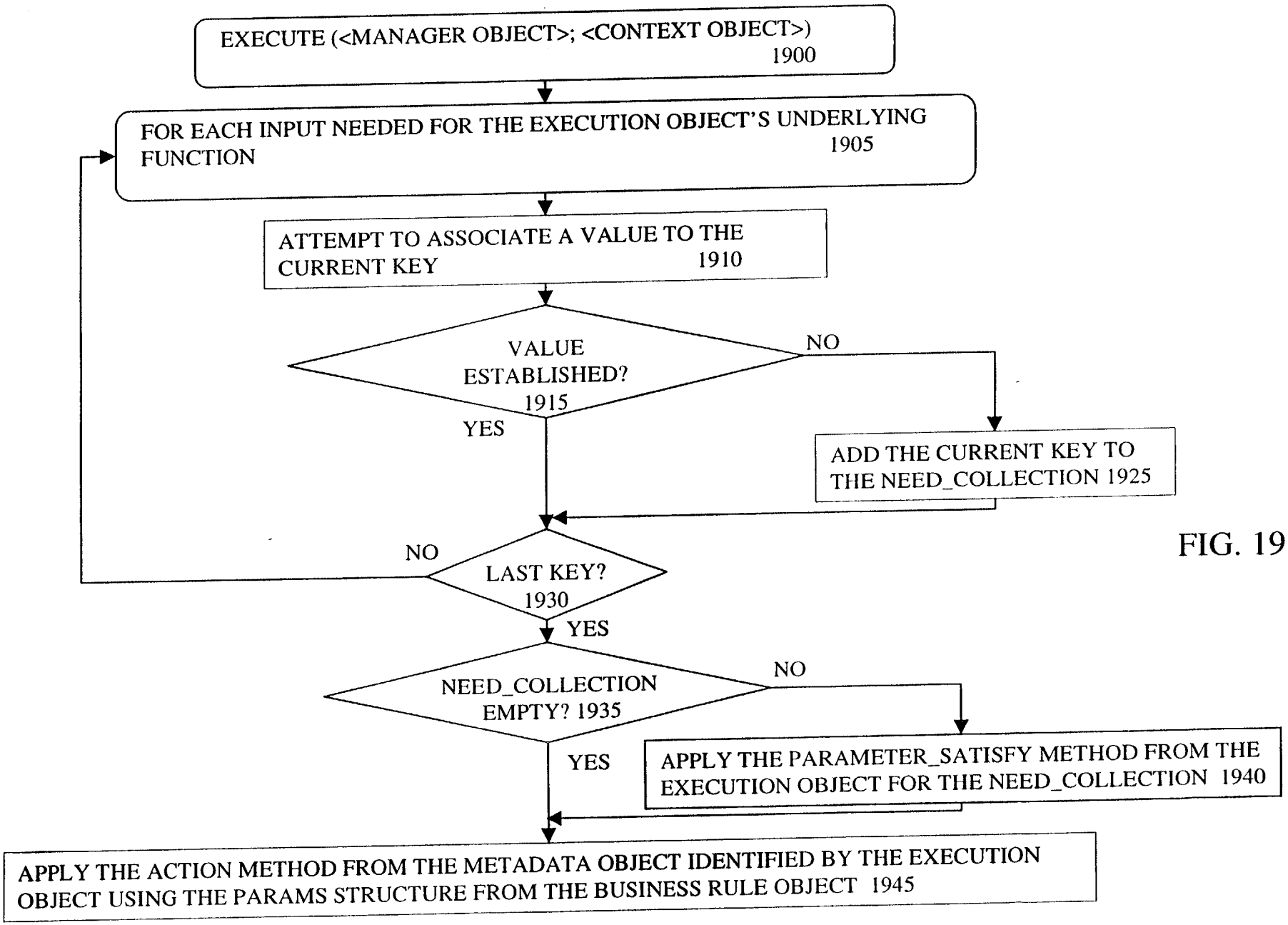
EXECUTE (<MANAGER OBJECT>; <CONTEXT OBJECT>)
1900
FOR EACH INPUT NEEDED FOR THE EXECUTION OBJECT'S UNDERLYING FUNCTION
1905
ATTEMPT TO ASSOCIATE A VALUE TO THE CURRENT KEY
1910
VALUE ESTABLISHED?
1915
NO
YES
ADD THE CURRENT KEY TO THE NEED_COLLECTION 1925
NO
LAST KEY?
1930
YES
NEED_COLLECTION EMPTY? 1935
NO
YES
APPLY THE PARAMETER_SATISFY METHOD FROM THE EXECUTION OBJECT FOR THE NEED_COLLECTION 1940
APPLY THE ACTION METHOD FROM THE METADATA OBJECT IDENTIFIED BY THE EXECUTION OBJECT USING THE PARAMS STRUCTURE FROM THE BUSINESS RULE OBJECT 1945

FIG. 19

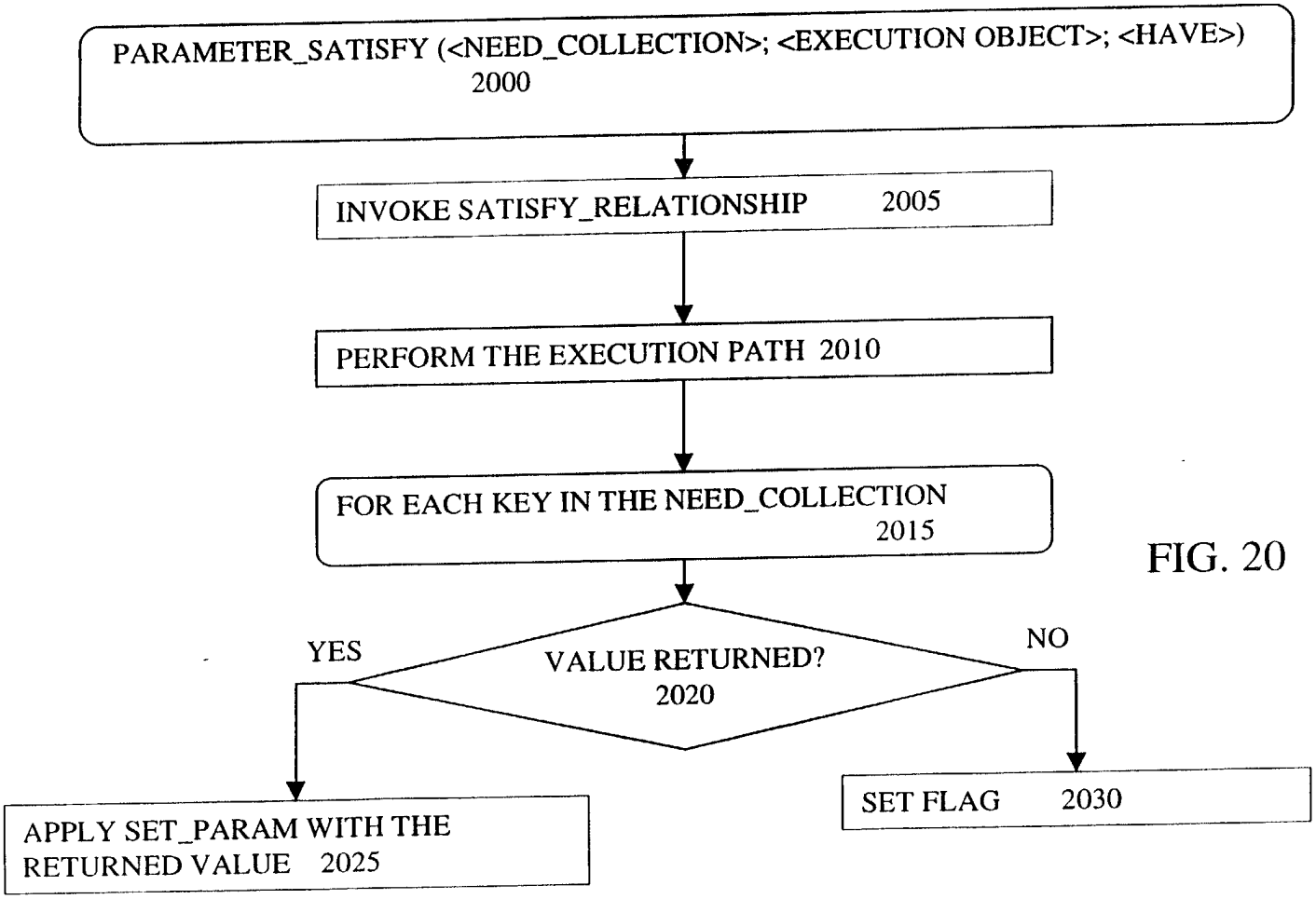
PARAMETER_SATISFY (<NEED_COLLECTION>; <EXECUTION OBJECT>; <HAVE>)
2000
INVOKE SATISFY_RELATIONSHIP 2005
PERFORM THE EXECUTION PATH 2010
FOR EACH KEY IN THE NEED_COLLECTION
2015
VALUE RETURNED?
2020
YES
NO
APPLY SET_PARAM WITH THE RETURNED VALUE 2025
SET FLAG 2030

FIG. 20

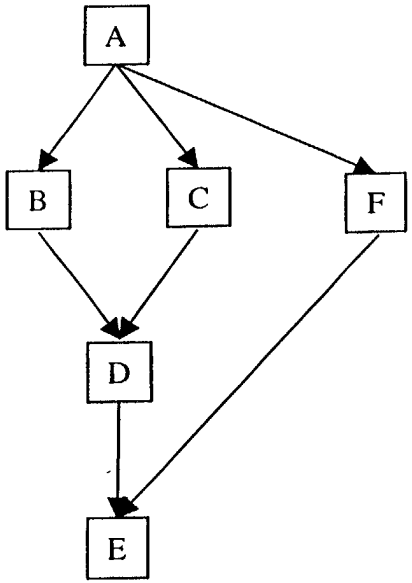

FIG. 21